MAUD : Un prototype de machine à dicter vocale

Dominique Fohr, Jean-Paul Haton, Jean-François Mari, Kamel Smaïli, Imed Zitouni

▶ **To cite this version:**

Dominique Fohr, Jean-Paul Haton, Jean-François Mari, Kamel Smaïli, Imed Zitouni. MAUD : Un prototype de machine à dicter vocale. none. Ressources et évaluation en ingénierie des langues, De Boeck, pp.315-328, 2000, universités francophones. inria-00099070

HAL Id: inria-00099070

https://inria.hal.science/inria-00099070

Submitted on 26 Sep 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

MAUD: Un prototype de machine à dicter vocale

D. Fohr, J.P. Haton, J.F. Mari, K. Smaïli, I. Zitouni

LORIA & INRIA-Lorraine
BP 239 54506 Vandeuvre-les-Nancy, France
{fohr, jph, jfmari, smaili, zitouni } @loria.fr

Résumé
La saisie vocale de textes et de données est un important champ d'application des systèmes de reconnaissance automatique de la parole, ainsi que l'un des bancs de test favoris de ces systèmes. L'équipe RFIA du LORIA développe actuellement une nouvelle version de son système de dictée vocale indépendant du locuteur pour grands vocabulaires, MAUD (Machine AUtomatique à Dicter), fondé essentiellement sur une approche stochastique. Le système MAUD se caractérise par l'utilisation de modèles et d'algorithmes originaux, notamment les modèles acoustiques de Markov du second ordre et l'algorithme de reconnaissance lexicale à optimalité locale. Des contraintes syntaxiques de plus en plus fortes sont appliquées pour pallier les insuffisances du modèle statistique. Dans ce but, nous utilisons une grammaire d'unification permettant de prendre en compte les phénomènes d'accord en genre et en nombre. Nous présentons dans la suite des résultats sur le corpus de test du projet AUPELF-UREF B 1.

1. Introduction

La machine à dicter constitue l'une des retombées prometteuses de la reconnaissance de la parole. L'équipe RFIA développe actuellement une nouvelle version de son système de dictée vocale indépendant du locuteur pour grands vocabulaires, MAUD (Machine AUtomatique à Dicter), fondé essentiellement sur une approche stochastique.

L'article est organisé comme suit. Le paragraphe 2 est consacré à une description de l'architecture logicielle de MAUD. Nous y présentons le rôle de chacune des quatre passes d'analyse. Nous détaillons plus particulièrement la deuxième passe consacrée à la construction d'un treillis de mots à l'aide de notre algorithme de fouille fondé sur des critères d'optimalité locale appelé algorithme de « Viterbi-bloc » (Kriouile, 1990). Le paragraphe 3 aborde deux aspects complémentaires de l'accès au lexique, le traitement des liaisons et plus généralement la prise en compte de connaissances phonologiques. La description du modèle de langage fait l'objet du paragraphe 4. Il s'agit d'un modèle hybride combinant un modèle positionnel probabiliste fondé sur des biclasses et des triclasses et un ensemble de règles grammaticales. Ces règles utilisent un formalisme de grammaire d'unification et permettent de lever certaines ambiguïtés.

Les méthodes conçues pour effectuer une classification manuelle ou automatique du lexique sont également présentées. Les premiers résultats sur le corpus de test sont donnés puis analysés au paragraphe 5. Le paragraphe 6 contient la conclusion de cet article. Nous donnons dans ce paragraphe quelques directions dans lesquelles nous comptons poursuivre notre travail d' élaboration de systèmes de dictée vocale.

2. Architecture du système

La nouvelle version du système MAUD que nous développons possède une architecture (Fig. 1) permettant d' élaborer une réponse à l' aide de plusieurs traitements sur le signal et sur ses représentations successives. Ce système fonctionne en quatre passes:

- détermination du genre du locuteur;
- élaboration d' un treillis de mots en utilisant des modèles de Markov cachés d' ordre 2 et un modèle de langage de type bigramme;
- recherche des N-meilleures phrases à l' aide d' un modèle de langage de type trigramme travaillant sur le treillis de mots;
- filtrage des phrases obtenues grâce à un modèle de langage fondé sur les biclasses et triclasses et d' une grammaire d' unification.

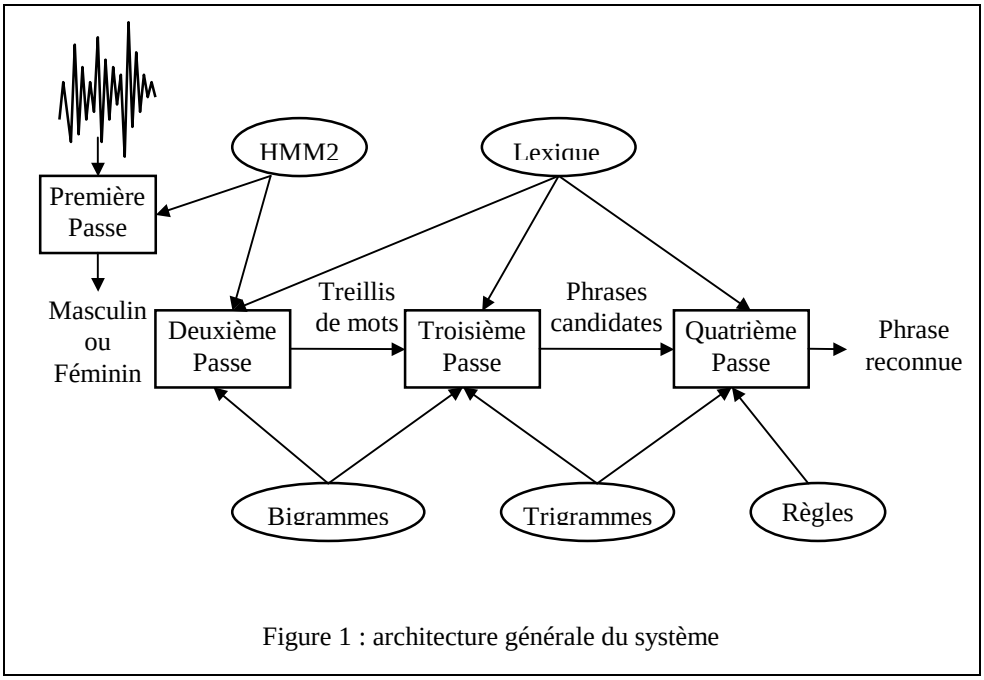

Figure 1 : architecture générale du système

2.1. Première passe : paramètrisation et détermination du genre du locuteur

Le signal est paramétré à l' aide de 12 coefficients cepstraux, ainsi que de leurs dérivées premières et secondes. Chaque trame est calculée toutes les 8 ms. Nous effectuons deux reconnaissances en parallèle, l' une avec 35 modèles de phonèmes non contextuels masculins, l' autre avec leurs homologues féminins. La meilleure vraisemblance détermine le genre du locuteur. Dans cette phase, le faisceau de recherche utilisé pendant la reconnaissance est très étroit afin d' accélérer le processus de sélection.

2.2. Deuxième passe : construction d'un treillis de mots

Cette passe a pour but d' obtenir, à partir du signal de parole, un treillis de mots en utilisant des modèles acoustiques de phonèmes contextuels masculins ou féminins suivant le résultat de la première passe ainsi que le lexique et un modèle de langage à contraintes locales (bigrammes). Chaque phonème en contexte (242 diphones masculins et 261 diphones féminins) est modélisé (Fohr, 1996) par un modèle de Markov caché continu du second ordre à trois états (HMM2) . Chaque HMM2 est appris à l' aide de BREF80 (Lamel, 1991).

Chaque mot du mot du lexique est représenté par un HMM2 obtenu par concaténation de HMM2 de diphones. Les différents allophones du même phonème sont représentés par des HMM2 partageant leurs 2 derniers états. Seul l' état de début diffère suivant l' allophone. Pour obtenir le treillis de mots nous utilisons un algorithme de type "Viterbi-bloc".

Contexte gauche trouvé	interprétation phonétique
i, e, j, k, g	antérieures
o, u, ã, õ, R, W	postériorisées
s, ch, t	dentales et chuintantes sourdes
@, y, O, l, z , d	dentales et chuintantes voisées
f, p	labiales sourdes
v, b	labiales sonores
in, m, n	nasales

Tableau 1: Liste des 7 diphones modélisant 1' allophone /a/

L'algorithme de Viterbi-bloc

L' algorithme de Viterbi-bloc que nous proposons se fonde sur une utilisation, pendant la reconnaissance, des critères basés sur une notion d' optimalité locale plutôt que globale. De telles contraintes ont été appliquées avec succès à la

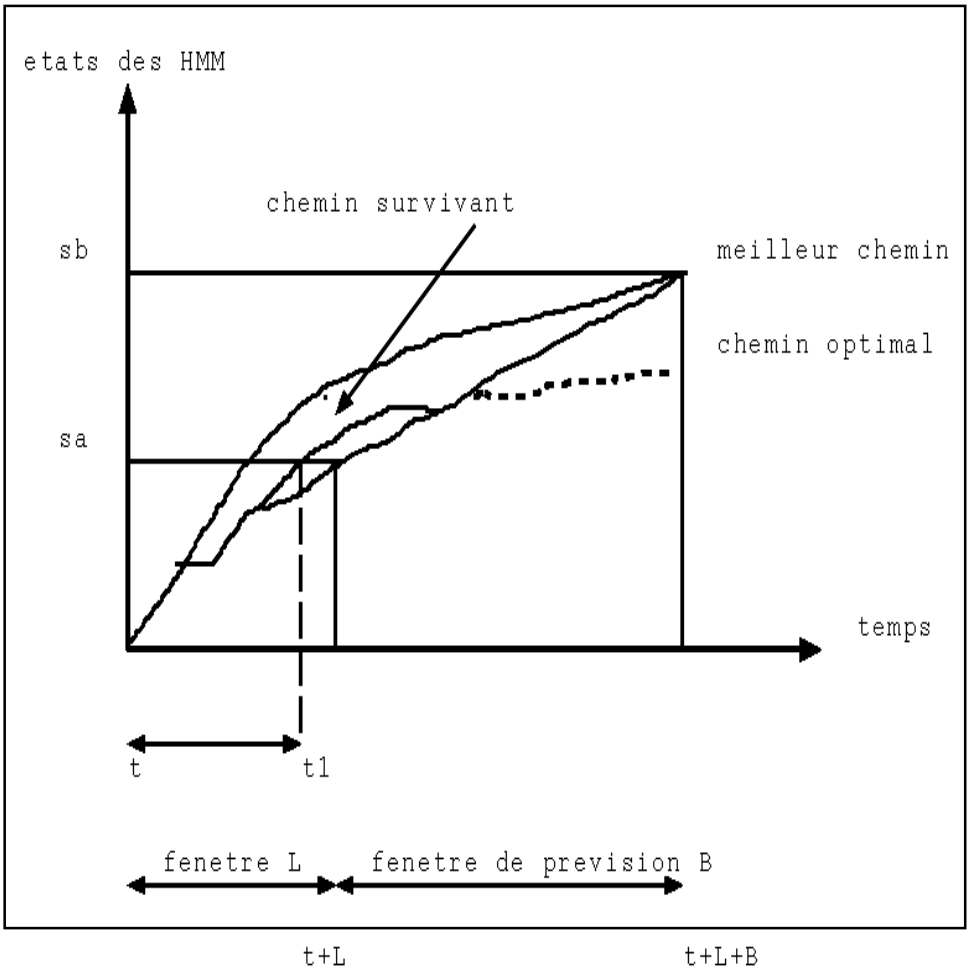

Figure 2 : construction du treillis de mots

reconnaissance de mots isolés par des méthodes de programmation dynamique (Haton, 1974), de traits acoustiques (Kriouile, 1990a ; Kriouile, 1990b) ou encore de phonèmes (Boulianne, 1994) dans le discours continu. Rappelons que l' algorithme de Viterbi, qui est un algorithme de programmation dynamique, construit et préserve à chaque instant, tous les chemins partiels s' achevant sur chaque état du HMM2 alors qu' un seul chemin, issu de l' état final associé à la dernière trame, sera utilisé pour retrouver la suite des états empruntés. Cette méthode devient vite inutilisable quand le nombre d' états augmente. En effet, les structures de données permettant les retours-arrières occupent alors une place trop importante. L' algorithme de Viterbbloc apporte une réponse à ce problème puisque la place mémoire nécessaire à son fonctionnement est indépendante de la longueur de la phrase et ne dépend que de la taille du bloc définissant la fenêtre de construction comme nous allons le voir. Un autre avantage de cet algorithme est d' effectuer une segmentation simultanément avec la reconnaissance et de donner une réponse en temps réel au fur et à mesure de la prononciation de la phrase, sans attendre la fin de celle-ci. Nous avons adapté cet algorithme à la recherche lexicale dans le contexte de la réalisation de la machine à dicter vocale.

L' algorithme construit un treillis de mots à l' aide de deux fenêtres: l' une, dite fenêtre de construction, de longueur L et l' autre dite fenêtre de prévision, de longueur L+B, dans laquelle un ou deux mots suivants sont recherchés (Fig. 2).

Supposons la reconnaissance effectuée jusqu' à la trame t. On appelle Sa l' état final du mot suivant pour lequel on cherche les frontières temporelles. Cet état est en fait l' état final du dernier phonème du mot suivant. On applique l' algorithme de Viterbi entre les trames [t, t+l], [t, t+2], ... [t, t+L]. Aux frontières de mots, le processus de construction utilise la grammaire bigramme fourni à la communauté AUPELF Bl. On poursuit le processus de construction à l' aide des mots suivants jusqu' à la trame t+L+B. Soit Sb un état possédant un score situé à l' intérieur du faisceau de recherche et qui s' aligne sur la trame t+L+B. Dans la fenêtre de prévision, nous venons de construire les chemins correspondants à deux mots supplémentaires.

A cet endroit, rien n' assure que l' on soit sur le chemin optimal donné par un algorithme de Viterbi classique. Mais, si la fenêtre de prévision est suffisamment grande -au plus, elle est égale à la portion restante de la phrase - on retrouvera le chemin optimal en reculant dans cette fenêtre car l' hypothèse expérimentale qui a motivé la spécification de l' algorithme de Viterbbloc est que le meilleur chemin s' achevant au point Sb, t+L+B est optimal entre t et t+L. Finalement, on se retrouve à l' instant t1 dans l' état Sa. t1 est l' extrémité recherchée du premier mot qui s' achève sur l' état Sa. Si ce mot n' a pas déjà été enregistré dans le treillis avec ces frontières temporelles, il est ajouté au treillis de mot. Le début de la fenêtre de

construction est déplacé sur la trame t1 et le processus de prévision est poursuivi à partir de la trame t1. Cet algorithme a été utilisé par l' équipe de l' INRS télécommunications et Bell-Northern pour étiqueter en phonèmes des phrases tirées de livres enregistrés sur des cassettes audiophoniques. La place mémoire nécessaire à l' exécution ne dépend que de la taille des fenêtres de construction et de prévision, ce qui rend ces algorithmes intéressants pour la reconnaissance de longues phrases; ce qui était très particulièrement le cas de l' étiquetage de livres enregistrés dans lesquels les phrases n' étaient pas artificiellement séparées par des pauses. Les auteurs ont déterminé expérimentalement les tailles des fenêtres de construction et de prévision pour conserver le chemin optimum dans la construction des chemins partiels d' étiquetage. Les durées de 0.6 secondes pour B et de 4 secondes pour L+B conviennent dans le cas du décodage acoustico-phonétique. La valeur de 0.6 secondes pour B est souvent inférieure à la durée moyenne d' un mot et nous avons remarqué que la majorité des chemins issus de Sb fusionnent pendant le retour-arrière entre les instants t+L et t+L+B.

A la fin de cette étape, on obtient une liste de mots candidats, avec pour chacun d' eux la probabilité d' alignement et ses instants de début et de fin.

Afin d' accélérer le processus de fouille, nous utilisons une recherche en faisceau qui détermine si un alignement s' achevant sur un mot doit être conservé. Nous calculons à chaque trame la valeur du seuil qui détermine l' abandon d' un chemin afin de ne conserver qu' un pourcentage fixé de l' ensemble des chemins. Deux valeurs influent sur le bon fonctionnement de cet algorithme: la largeur du faisceau de fouille et la longueur de la fenêtre de prévision B. Une valeur nulle pour B oblige à garder un faisceau de largeur maximale pour garder l' optimalité de la fouille. Inversement, une valeur de B égale à la longueur de la plus grande phrase se contentera d' un faisceau de largeur nulle.

Résultats

Nous avons effectué différentes expériences sur le corpus de développement afin de déterminer expérimentalement les valeurs de L, B ainsi que la largeur du faisceau de fouille. A l' heure actuelle, les fenêtres de construction et de prévision ont des tailles variables et correspondent aux durée des mots issus de t et s' achevant respectivement en Sa et Sb. Ceci revient à enregistrer l' avant dernier mot de chaque chemin partiel qui s' achève en Sb à l' instant t+L+B. Nous avons constaté que le nombre de mots qui s' achèvent en Sb à l' instant t+L+B est 40 fois plus important que le nombre de mots qui s' achèvent en Sa à l' instant t1.

2.3. Troisième passe : construction des N-meilleures phrases

Au cours de cette troisième étape, on cherche à construire les N-meilleures phrases à partir du treillis de mots obtenus à l' étape précédente. La méthode utilisée est une recherche en faisceau de type A* avec un modèle de langage trigramme, mais sans utiliser les modèles acoustiques, puisqu' on dispose de la probabilité d' alignement donnée par la passe précédente. Le résultat de cette étape est une liste de phrases ordonnées suivant un score qui combine la probabilité de recalage acoustique et le modèle de langage trigramme. Les résultats donnés au paragraphe 5 sont calculés à partir de la meilleure de ces phrases.

2.4. Quatrième passe : filtrage de phrases

A la différence des deux premières étapes où seules des contraintes très locales sont utilisées, lors de cette dernière passe, il s'agit d' ajouter des contraintes à plus long terme. Les contraintes linguistiques utilisées et les règles grammaticales sont détaillées dans la partie consacrée au modèle grammatical formel. La première phrase respectant ces nouvelles contraintes est donc la phrase reconnue par le système. Dans la version actuelle du système, la quatrième passe n' a pas encore été introduite.

3. Accès lexical, liaison et phonologie

Un des problèmes inhérent à la reconnaissance de la parole continue est la possibilité de faire des liaisons entre les mots. Pour prendre en compte ce phénomène, nous traitons cette information au niveau du modèle markovien. Comme nous l' avons déjà mentionné, nous représentons chaque mot par un modèle de Markov constitué de la connexion des différents HMM2 de diphones. Quand un mot peut être prononcé en liaison avec le mot qui le suit, nous ajoutons à la chaîne de phonèmes qui le compose le phonème correspondant à la consonne latente. Par exemple, le mot "irons" sera représenté par la suite de phonèmes /irõz/ et nous alignerons /irõ/ si le mot suivant commence par une consonne. Si le mot suivant commence par une voyelle, nous alignerons /irõ/ et /irõz/ et nous ne garderons que le meilleur alignement. Dans un soucis de simplification, nous considérons que les liaisons sont facultatives. Pour savoir si un mot est susceptible de provoquer une liaison, nous utilisons l' information contenue dans le lexique BDLEX développé à Toulouse par l' IRIT.

Quand un mot peut se prononcer de plusieurs manières (par exemple le mot "médecin"), nous avons choisi de créer un modèle de Markov par prononciation. Quand un mot peut se terminer par un "e muet", nous alignons systématiquement les deux prononciations (avec et sans "e muet") et nous ne gardons que le meilleur

alignement. En revanche, nous ne traitons pas les assimilations de sonorité quand le "e muet" (noté @) disparaît, comme dans "banque de France". Nous essayerons d' aligner "bãk@d@fRãs" ou "bãkd@frãs" mais pas "bãgd@frãs".

4. Le modèle de langage

Malgré l' importante avancée des modèles de langage statistiques dans les systèmes de reconnaissance de la parole, ces modèles restent néanmoins limités dans leur pouvoir de filtrage des hypothèses acoustiques et suffisamment permissifs au niveau prédictif. L' intérêt des modèles de langage statistiques est leur relative facilité de mise en œuvre. La construction de ces modèles se matérialise par un apprentissage simple des paramètres servant, par la suite, à modéliser le comportement de l' application au niveau langagier.

Pour que ces modèles soient efficaces, il faut disposer de corpus de taille importante pour pouvoir estimer les paramètres d' une manière correcte. Ces corpus ne sont pas toujours disponibles et, le cas échéant, ils ne peuvent servir qu' à la conception de modèles positionnels du langage. Autrement dit, les structures grammaticales prises en compte par ces modèles sont composées de deux ou trois mots[1]. Il se trouve que dans une langue, il y a souvent des mots qui font référence à d' autres qui ne se trouvent pas forcément deux ou trois mots avant le mot courant. Ces références peuvent être de nature sémantique ou purement syntaxique. Ainsi les phénomènes d' accord en genre et en nombre en sont un bon exemple. Dans la phrase: *George et Marie, tous deux cousins de Gilles, sont des amis de Paul,* l' auxiliaire *être* s' accorde avec le sujet qui se trouve 6 mots avant. Il n' est donc pas possible à un modèle de langage statistique positionnel de prendre en compte des phénomènes d' accord de ce type. C' est pour cette raison que nous défendons depuis quelques années (Smaïli, 1991) l' approche hybride qui consiste à combiner les avantages d' un modèle statistique avec ceux d' un modèle grammatical classique.

4.1. Le modèle statistique de langage

Le modèle statistique de langage de MAUD est fondé sur un apprentissage semi-automatique de suites de biclasses et triclasses effectué sur le corpus de textes de 40 millions de mots fourni par l' AUPELF. Cette opération d' étiquetage de textes nécessite l' utilisation d' un lexique. Le lexique de MAUD n' est pas constitué seulement d' une liste de mots, mais d' un ensemble d' informations complexes

[1] Dans certains modèles, ces structures atteignent une taille de 5.

comportant deux parties: infra-lexicale et supra-lexicale. Parmi les informations nécessaires à l' opération d' étiquetage de textes, nous citons les classes syntaxico-sémantiques. Le dictionnaire de MAUD est composé d' un sous-ensemble du dictionnaire de BDLEX auquel ont été ajoutées des informations fréquentielles et syntaxico-sémantiques. Dans une première expérience, nous avons développé un jeu de classification de 201 classes, construit manuellement, à l' aide de propriétés syntaxiques prédéfinies (Smaïli, 1993).

4.2. Le modèle grammatical formel

Etant données les limites des modèles statistiques, il est nécessaire si l' on veut réduire le nombre d' hypothèses proposées par les niveaux inférieurs d' un système de reconnaissance de la parole d' ajouter explicitement dans le modèle des connaissances linguistiques. Ces connaissances vont permettre de préciser comment l' information linguistique associée à chacun des éléments constitutifs d' une phrase se combine. Pour ce faire, nous avons écrit quelques règles grammaticales du français modélisées par une grammaire d' unification.

Rappelons que les grammaires d' unification sont fondées sur la notion de structures d' information ou de trait. Chaque structure d' information est composée d' un ensemble de paires<*attribut, valeur*> telles que chaque attribut ne peut prendre qu' une seule valeur. Ainsi, dans l' exemple ci-dessous, nous disposons d' une structure à trois attributs: catégorie*(CAT),* personne *(PERS)* et genre *(GENR)* dont les valeurs sont *sn,* 3, et *F* respectivement.

[CAT: sn, PERS: 3, *CENR: F]*

Un des intérêts de ces structures est qu'il est possible de les ordonner selon leur degré d' information. En effet, une structure d' information relative à un objet linguistique est souvent partielle. Dans les deux structures ci-dessous, l' information contenue dans ST2 est plus générale que celle de ST1.

[CAT: V] (ST1)

[CAT: V, RACINE: courir] (ST2)

On dira qu' une structure de traits (ST1) subsume une autre structure (ST2) si ST2 est au moins aussi informative que ST1 (on dira également que ST2 étend ST1). Les grammaires d' unification se basent sur la subsomption pour unifier deux structures d' information. L' opération d' unification combine l' information contenue

dans deux structures de traits à condition qu' elles soient compatibles. Ainsi, une règle du type S → GN GV peut être exprimée dans le formalisme d' une grammaire d' unification par la règle et les relations d' unification suivantes:

$X_0 \rightarrow X_1\ X_2$ et

$CAT_0 = S$
$CAT_1 = GN$
$CAT_2 = GV$
$NBP_0 = NBP_1 = NBP_2$
$FORMV_0 = FORM_2$

où NBP désigne le nombre et la personne du sujet. Le trait NBP peut prendre six valeurs: 1s, 2s, 3s, 1p, 2p et 3p. FORMV désigne la forme du verbe.

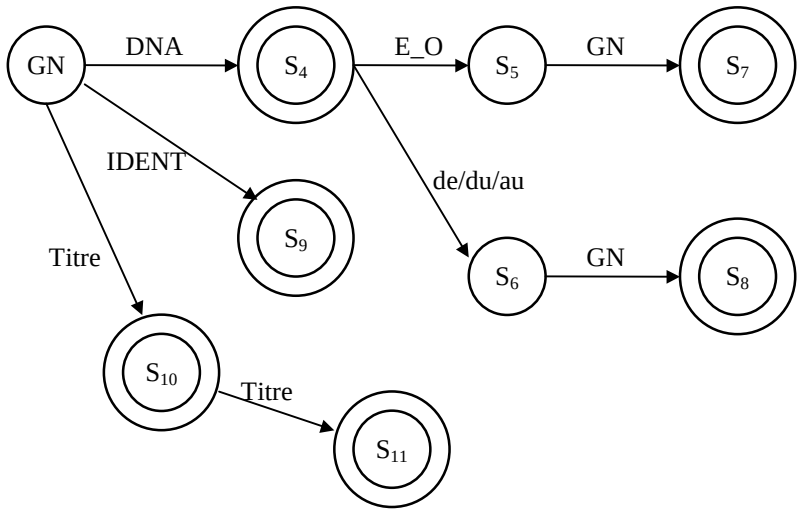

Fig. 3: Exemples de règles grammaticales utilisées par MAUD

L' interprétation de la règle ci-dessus se fait de la manière suivante: Le constituant X_0 peut être construit à partir de X_1 et X_2 si la catégorie de X_0 est S. la catégorie de X_1 est GN, la catégorie de X_2 est GV, les valeurs du trait NBP sont les mêmes pour les trois constituants et les valeurs de FORMV des constituants X_0 et X_2 sont identiques.

Dans la prochaine version de MAUD, nous implanterons une grammaire d' unification pour laquelle un certain nombre de traits ont été définis. Pour des facilités de présentation, nous donnerons dans ce qui suit quelques exemples de règles présentées sous forme d' ATN. Ces ATN utilisent des non-terminaux qui renvoient vers d' autres ATN et des terminaux qui peuvent représenter soit des mots du lexique soit des classes syntaxico-sémantiques qui ont été définies dans la phase de classification (Fig. 3).

Les structures de traits associées à ces automates sont supportées par des procédures qui sont déclenchées à chaque rencontre d' un état final dans l' analyse de la phrase. Les états finaux sont représentés dans les automates par des doubles cercles concentriques.

4.3. Stratégie de filtrage du modèle de langage

Dans le cadre du projet AUPELF, le modèle de langage nous sert, entre autres, à filtrer les hypothèses de phrases proposées par les niveaux inférieurs de MAUD. En effet, une fois les phrases constituées, il restera à filtrer l' ensemble des phrases qui ont été reconnues et qui sont passées à travers le filtre du modèle trigramme. Pour ce faire, un étiquetage syntaxique des phrases sera nécessaire afin de résoudre syntaxiquement les mots qui composent ces phrases. Une fois cette opération effectuée, un premier filtrage permettra de garder les N meilleures phrases selon la valeur Q du modèle stochastique que nous utilisons. La valeur de Q est donnée par la formule ci-dessous:

$$Q = \prod_{i=3}^{n} \left[P\langle C_i \,|\, w_i \rangle \right] \times \left[\lambda_1 \cdot P(C_i) + \lambda_2 \cdot P\langle C_i \,|\, C_{i-1} \rangle + \lambda_3 \cdot P\langle C_i \,|\, C_{i-1} C_{i-2} \rangle + \lambda_4 \right]$$

Ensuite, la grammaire d' unification prend le relais pour filtrer les phrases qui sont passées à travers ce filtre stochastique. La grammaire d' unification n' éliminera que les phrases dont la structure est représentée par un de ses automates. Autrement dit, toutes les phrases qui ne peuvent être analysées par la grammaire d' unification feront partie des propositions du système MAUD.

5. Résultats

Nous présentons ici des résultats obtenus sur 20 locuteurs représentant 299 phrases issues du corpus de test.

Les taux de reconnaissance, présentés dans le tableau 2, sont donnés au niveau du mot.

Nombre total de mots	9061
Mots corrects	70.3 %
Mots substitués	21.5 %
Mots omis	8.2 %
Mots insérés	2.3 %

Tableau 2: taux de reconnaissance obtenu sur le corpus de test

Ces taux sont bien inférieurs aux résultats publiés dans ce genre d' application. Plusieurs explications peuvent être avancées. La principale raison est la jeunesse du système. Il y a un an, notre équipe commençait à étiqueter BREF80 en mots et phonèmes afin de faire converger les HMM2 de phonèmes non-contextuels. A l' heure actuelle, nous n' avons implanté que des allophones en contexte gauche; la construction de triphones généralisés constitue notre prochain objectif. La spécification du lexique devra être améliorée afin de tenir compte des phénomènes de coarticulation et d' assimilation de telle sorte que la définition d' un mot ne soit plus simplement une suite de phonèmes mais autorise plusieurs variantes suivant les prononciations admissibles. De plus, aucun algorithme n' a été utilisé pour palier le problème des mots prononcés n' appartenant pas au lexique. Enfin, en ce qui concerne le modèle de langage, nous avons simplement implémenté les modèles bigrammes et trigrammes qui nous ont été fournis sans utiliser le modèle détaillé au paragraphe 4. D' une façon générale, l' importance de la tâche de programmation ainsi que la relative complexité de l' application a rendu laborieuse la mise au point des programmes; ainsi les résultats obtenus sur le corpus de test ont été rendus bien avant d' avoir réussi à exploiter entièrement le corpus de développement. Beaucoup de paramètres du système ne sont pas bien réglés si bien que les résultats fournis ne reflètent pas les possibilités réelles des méthodes originales que nous avons choisies et nous comptons bien améliorer notre système dans les mois à venir.

6. Conclusions

Nous avons présenté dans cet article différents aspects originaux de notre système de dictée vocale en parole continue MAUD. Ce système se fonde essentiellement sur une approche stochastique, depuis le niveau acoustique jusqu' à celui de la phrase, tout en faisant appel à des connaissances de diverses natures pour compléter les modèles stochastiques.

Le système MAUD se caractérise par l' utilisation de modèles et algorithmes

originaux, notamment les modèles acoustiques de Markov du second ordre et l' algorithme de reconnaissance lexicale Viterbi-bloc. Des contraintes syntaxiques de plus en plus fortes sont appliquées pour pallier les insuffisances du modèle statistique. Dans ce but, nous utilisons une grammaire d' unification permettant de prendre en compte les phénomènes d' accord en genre et en nombre.

Dans le cadre de l' action AUPELF-UREF d' évaluation des systèmes francophones de dictée vocale, nos premiers résultats expérimentaux complets sont quelque peu décevants et témoignent d' une grande jeunesse du système. Parmi les extensions prévues de nos travaux, on peut citer la modélisation probabiliste de triphones généralisés, l' amélioration des modèles statistiques de langage ainsi que la modélisation des connaissances phonologiques. On peut citer également l' apprentissage automatique d' une grammaire d' unification qui s' inspire des techniques d' inférence grammaticale. Ces travaux ont pour objectif d' améliorer les performances du système qui reste bien en deçà des possibilités des algorithmes originaux employés.

Remerciements

Nous tenons à remercier le centre Charles Hermite de calcul à haute performance et tout particulièrement Alain Filbois pour les facilités d' accès accordées durant la phase de test.

Références

André-Olbrecht, R. (1988). A New Statistical approach for the automatic segmentation of continuous speech signals, IEEE Transactions on Acoustics Speech Signal processing, 36 (1).

Boulianne, G. & Kenny, P. & Lening, M. & O' Shaughessy, D. & Mermelstein, P. (1994). Books on tape as training data for continuous speech recognition, Speech communication, 14:61.

Fohr, D. & Mari, J.-F. & Haton, J.-P. (1996). Utilisation de modèles de Markov pour l' étiquetage automatique et la reconnaissance de BREF80. XXIème journées d' étude sur la parole (pp. 339 - 342). Avignon.

Haton, J.P. (1974). Contribution à l' analyse, la paramétrisation et la reconnaissance automatique de la parole. Thèse de doctorat. Université de Nancy 1.

Kriouile, A. & Mari, J.F. & Haton, J.P. (1990). L' algorithme Viterbi-bloc pour la reconnaissance de la parole continue. XVIII journées d' étude sur la parole, pp 207-210.

Lamel, L. & Gauvain, J.L. & Eskénasi, M. (1991). BREF, a Large Vocabulary

Spoken Corpus for French, Eurospeech
Mari, J.-F & Haton, J.-P. & Kriouile (1997). Automatic Word Recognition Based on Second-Order Hidden Markov Models. In IEEE Transactions on Speech and Audio Processing, 2(1), 22 - 25.
Smaïli, K. (1991). Conception et réalisation d' une machine à dicter à entrée vocale destinée aux grands vocabulaires: le système MAUD. Thèse de doctorat de l' université de Nancy 1.
Smaïli, K. & Charpillet, F. & Haton, J.-P. (1993). MAUD: une interface vocale pour la saisie de textes lus, 2èmes journées internationales Interfaces des mondes réels et virtuels.